In the name of Allah, the Most Gracious, the Most Merciful

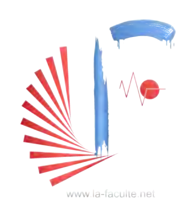




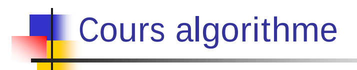










# Cours algorithme



## Introduction à l'Algorithmique

- **Un algorithme, c'est une suite d'instructions, qui une fois exécutée correctement, conduit à un résultat donné**
- l'algorithmique exprime les instructions résolvant un problème donné **indépendamment des particularités de tel ou tel langage** :
  - structure logique X langage de programmation



- Pour fonctionner, **un algorithme doit donc contenir uniquement des instructions compréhensibles par celui qui devra l'exécuter**
- **la vérification méthodique, pas à pas, de chacun de vos algorithmes représente plus de la moitié du travail à accomplir... et le gage de vos progrès.**



## Exemple d'un algorithme simple: Préparation d'un litre de glace

Entrée

- ½ litre de lait
- 6 œufs
- 200 grammes de sucres glaces
- 2 cuillères à café de café soluble

Corps de l'algoithme

- 1- faire bouillir le lait
- 2- battre les jaunes d'œufs avec le sucre
- 3- verser dessus le lait bouillant en remuant avec une spatule
- 4- laisser congeler dans le réfrigérateur



## Définition d'un algorithme

Un algorithme est une séquence (suite) d'actions primitives, qui exécutés par un processeur bien défini réalisera un travail bien précis (demandé)

Propriétés:

Il doit tenir compte de tous les cas possibles.

Il traite le cas général et les cas particuliers

Il contient toujours un nombre fini d'actions

Il est en général répétitif ( il contient un traitement qui se répéte )



## Exemple d'un algorithme avec répétitions: Appel téléphonique d'un correspondant en Algérie

Début

1. Appeler la personne
   - décrocher le combiné
   - attendre la tonalité
   - composer le numéro de votre correspondant (9 ou 10 chiffres selon le cas)
2. Si le correspondant réponds alors parler un peu et ensuite aller vers 4
3. Si correspondant ne réponds pas alors
   Tant que le correspondant ne réponds pas:
   Si nombre de rappel dépasse 5 alors aller vers 4
   sinon refaire les instructions du bloc 1
4. Raccrocher le combiné

Fin



## Principales constituantes d'un algorithme

- Variables
- Affectations
- Expressions
- Lecture & écriture
- Structures de contrôle



## Les Variables

- Dans un programme informatique, on va avoir en permanence besoin de stocker provisoirement des valeurs de types différents: utilisation de variables
- une variable est une **boîte**, que le programme (l'ordinateur) va repérer par une **étiquette**. Pour avoir accès au contenu de la boîte, il suffit de la désigner par son étiquette.



## Déclaration des variables

- **Il s'agit de créer la boîte et de lui coller une étiquette**.
- Le **nom** de la variable (l'étiquette de la boîte) obéit à des impératifs changeant selon les langages:
  - Un nom de variable correct commence impérativement par une lettre.
  - comporte des lettres et des chiffres, mais qui exclut la plupart des signes de ponctuation, en particulier les espaces.



- Le **type** de la boite précise ce que l'on voudra mettre dedans, car de cela dépendent la **taille** de la boîte
  - Types numériques
  - Type alphanumérique
  - Type booléen



## Types numérique

Une variable destinée à recevoir des nombres.

| Type Numérique | Plage |
|---|---|
| Byte (octet) | 0 à 255 |
| Entier simple | -32 768 à 32 767 |
| Entier long | -2 147 483 648 à 2 147 483 647 |
| Réel simple | $-3,40 \times 10^{38}$ à $-1,40 \times 10^{45}$ pour les valeurs négatives<br>$1,40 \times 10^{-45}$ à $3,40 \times 10^{38}$ pour les valeurs positives |
| Réel double | $1,79 \times 10^{308}$ à $-4,94 \times 10^{-324}$ pour les valeurs négatives<br>$4,94 \times 10^{-324}$ à $1,79 \times 10^{308}$ pour les valeurs positives |

- **Définir le type en fonction des besoins**



## Pseudo-code

- **Variable** g **en Numérique**

ou encore

- **Variables** PrixHT, TauxTVA, PrixTTC **en Numérique**



## Type alphanumérique

- également appelé **type caractère**, **type chaîne** ou en anglais, le **type string**
  - lettres, signes de ponctuation, espaces, ou même de chiffres.
- Le nombre maximal de caractères pouvant être stockés dans une seule variable **string** dépend du langage utilisé.



- une chaîne de caractères est toujours notée entre guillemets
  - éviter la confusion:
    - entre des nombres et des suites de caractères chiffres.
    - entre le nom d'une variable et son contenu



## Type booléen

- uniquement les valeurs logiques VRAI et FAUX: (TRUE et FALSE) ou des nombres (0 et 1)
- le type booléen est très économique en termes de place mémoire occupée, puisque pour stocker une telle information binaire, un seul bit suffit.



## L'instruction d'affectation

- affecter une variable c'est lui attribuer une valeur i.e mettre un contenu dans la boite
- En pseudo-code, l'instruction d'affectation se note avec le signe ←
  - Toto ← 24 //*Attribue la valeur 24 à la variable Toto*
- Comptabilité entre le contenant et le contenu
- Une variable désigné doit être au préalable déclarée



## Exemple d'affectation

- Toto ← 3
- Tutu ← Toto
- Tutu ← Toto + 4
- Tutu ← Tutu + 1
- Quel est la valeur finale de la variable Tutu ?



- **Variable** A **en Numérique**
  **Début**
  A ← 34
  A ← 12
  **Fin**
- **Variable** A **en Numérique**
  **Début**
  A ← 12
  A ← 34
  **Fin**

L'ordre dans lequel les instructions sont écrites va jouer un rôle essentiel dans le résultat final.



- **Exercice 1.1**
- Quelles seront les valeurs des variables A et B après exécution des instructions suivantes ?
- **Variables** A, B **en Entier**
  **Début**
  A ← 1
  B ← A + 3
  A ← 3
  **Fin**

**Fin**

- corrigé -



## Corrigé

- Après La valeur des variables est :
  A ← 1 A = 1 B = ?
  B ← A + 3 A = 1 B = 4
  A ← 3 **A = 3** **B = 4**



- **Exercice 1.2**
- Quelles seront les valeurs des variables A, B et C après exécution des instructions suivantes ?
- **Variables** A, B, C **en Entier**
  **Début**
  A ← 5
  B ← 3
  C ← A + B
  A ← 2
  C ← B – A
  **Fin**
- corrigé -



## Corrigé

- Après La valeur des variables est :
  A ← 5 A = 5 B
  = ? C = ?
  B ← 3 A = 5 B =
  3 C = ?
  C ← A + B A = 5 B =
  3 C = 8
  A ← 2 A = 2 B =
  3 C = 8
  C ← B – A **A = 2** **B =**



- **Exercice 1.3**
- Quelles seront les valeurs des variables A et B après exécution des instructions suivantes ?
- **Variables** A, B **en Entier**
  **Début**
  A ← 5
  B ← A + 4
  A ← A + 1
  B ← A – 4
  **Fin**
- corrigé -



## Corrigé

- Après La valeur des variables est :
  A ← 5 A = 5 B = ?
  B ← A + 4 A = 5 B = 9
  A ← A + 1 A = 6 B = 9
  B ← A – 4 **A = 6** **B = 2**



- **Exercice 1.4**
- Quelles seront les valeurs des variables A, B et C après exécution des instructions suivantes ?
- **Variables** A, B, C **en Entier**
  **Début**
  A ← 3
  B ← 10
  C ← A + B
  B ← A + B
  A ← C
  **Fin**
- corrigé -



- Après La valeur des variables est :
  A ← 3 A = 3 B = ? C = ?
  B ← 10 A = 3 B = 10 C = ?
  C ← A + B A = 3 B = 10 C = 13
  B ← A + B A = 3 B = 13 C = 13
  A ← C **A = 13** **B = 13** **C = 13**



- **Exercice 1.5**
- Quelles seront les valeurs des variables A et B après exécution des instructions suivantes ?
- **Variables** A, B **en Entier**
  **Début**
  A ← 5
  B ← 2
  A ← B
  B ← A
  **Fin**
- Moralité : les deux dernières instructions permettent-elles d'échanger les deux valeurs de B et A ? Si l'on inverse les deux dernières instructions, cela change-t-il quelque chose ?
- corrigé -



## Corrigé

- Après La valeur des variables est :
  A ← 5 A = 5 B = ?
  B ← 2 A = 5 B = 2
  A ← B A = 2 B = 2
  B ← A **A = 2** **B = 2**
- Les deux dernières instructions ne permettent donc pas d'échanger les deux valeurs de B et A, puisque l'une des deux valeurs (celle de A) est ici écrasée. Si l'on inverse les deux dernières instructions, cela ne changera rien du tout, hormis le fait que cette fois c'est la valeur de B qui sera écrasée.



- **Exercice 1.6**
- Plus difficile, mais c'est un classique absolu, qu'il faut absolument maîtriser : écrire un algorithme permettant d'échanger les valeurs de deux variables A et B, et ce quel que soit leur contenu préalable.
- corrigé -



- **Début**
  …
  C ← A
  A ← B
  B ← C
  **Fin**
- On est obligé de passer par une variable dite temporaire (la variable C).



- **Exercice 1.7**
- Une variante du précédent : on dispose de trois variables A, B et C. Ecrivez un algorithme transférant à B la valeur de A, à C la valeur de B et à A la valeur de C (toujours quels que soient les contenus préalables de ces variables).
- corrigé -



- **Début**
  ...
  D ← C
  C ← B
  B ← A
  A ← D
  **Fin**
- En fait, quel que soit le nombre de variables, une seule variable temporaire suffit...



# Expressions et opérateurs

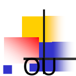
- Une expression est un ensemble d'opérandes, reliées par des opérateurs, et équivalent à une seule valeur
- Exemple d'expression (opérandes numériques):
  ***7; 5+4; 123-45+844;Toto-12+5-Riri***;
- Opérandes: numérique, alphanumérique, booléen...



- Opérateurs numériques
  - + : addition
  - - : soustraction
  - * : multiplication
  - / : division
  - ^ : puissance
  - ...
- Priorité des opérateurs: * et / sur + et -
  - *12 * 3 + 5 ≠ 12*(3+5)*



- Opérateur alphanumérique : &
  - Variables A, B, C en Caractère
    Début
    A ← "Gloubi"
    B ← "Boulga"
    C ← A & B
    Fin
- Opérateurs logiques (ou booléens)
  - Ou, ET, NON, ...



- OU

| A | B | A ou B |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

- ET

| A | B | A ET B |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

- NON

| A | | NON A |
|---|---|---|
| 0 | | 1 |
| 1 | | 0 |



- **Exercice 1.8**
- Que produit l'algorithme suivant ?
- **Variables** A, B, C **en Caractères**
  **Début**
  A ← "423"
  B ← "12"
  C ← A + B
  **Fin**
- corrigé -



- Il ne peut produire qu'une erreur d'exécution, puisqu'on ne peut pas additionner des caractères.



- **Exercice 1.9**
- Que produit l'algorithme suivant ?
- **Variables** A, B **en Caractères**
  **Début**
  A ← « Télé"
  B ← « vision"
  C ← A & B



- A la fin de l'algorithme « Télévision".



- **Exercice 1.9**
- Que produit l'algorithme suivant ?
- **Variables** A, B **en Caractères**
  **Début**
  A ← "423"
  B ← "12"
  C ← A & B



- A la fin de l'algorithme "42312".



## Lecture et Ecriture



- Dans un sens, ces instructions permettent à l'utilisateur de rentrer des valeurs au clavier pour qu'elles soient utilisées par le programme. Cette opération est la **lecture**.
- Dans l'autre sens, d'autres instructions permettent au programme de communiquer des valeurs à l'utilisateur en les affichant à l'écran. Cette opération est **l'écriture**.



## Les instructions de lecture et d'écriture

**Lire Titi**

- Dès que le programme rencontre une instruction Lire, l'exécution s'interrompt, attendant la frappe d'une valeur au clavier
- aussitôt que la touche Entrée (Enter) a été frappée, l'exécution reprend



Dans le sens inverse, pour écrire quelque chose en sortie :

**Ecrire** Toto

D'habitude l'affichage suit un certains formatage



## Exercices

- Quel résultat produit le programme suivant ?
- Variables val, double numériques
  Début
  Val ← 231
  Double ← Val * 2
  Ecrire Val
  Ecrire Double
  Fin

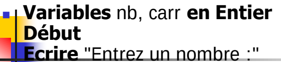


## Corrigé

- On verra apparaître à l'écran 231, puis 462 (qui vaut 231 * 2)



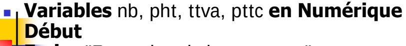
- Ecrire un programme qui demande un nombre à l'utilisateur, puis qui calcule et affiche le carré de ce nombre.



- **Variables** nb, carr **en Entier**
  **Début**
  **Ecrire** "Entrez un nombre :"
  **Lire** nb
  carr ← nb * nb
  **Ecrire** "Son carré est : ", carr
  **Fin**
- En fait, on pourrait tout aussi bien économiser la variable carr en remplaçant les deux avant-dernières lignes par :
- **Ecrire** "Son carré est : ", nb*nb
- C'est une question de style ; dans un cas, on privilégie la lisibilité de l'algorithme, dans l'autre, on privilégie l'économie d'une variable.



- Ecrire un programme qui lit le prix HT d'un article, le nombre d'articles et le taux de TVA, et qui fournit le prix total TTC correspondant. Faire en sorte que des libellés apparaissent clairement.



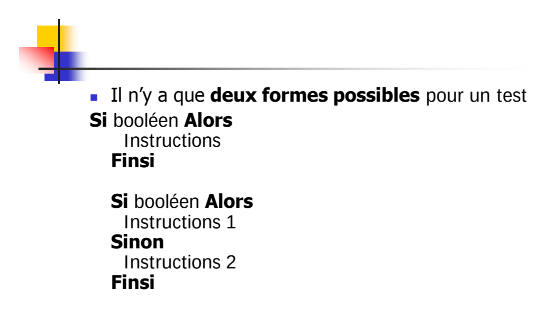
- **Variables** nb, pht, ttva, pttc **en Numérique**
  **Début**
  **Ecrire** "Entrez le prix hors taxes :"
  **Lire** pht
  **Ecrire** "Entrez le nombre d'articles :"
  **Lire** nb
  **Ecrire** "Entrez le taux de TVA :"
  **Lire** ttva
  pttc ← nb * pht * (1 + ttva)
  **Ecrire** "Le prix toutes taxes est : ", pttc
  **Fin**
- Là aussi, on pourrait squeezer une variable et une ligne en écrivant directement. :
- **Ecrire** "Le prix toutes taxes est : ", nb * pht * (1 + ttva)
- C'est plus rapide, plus léger en mémoire, mais un peu plus difficile à relire (et à écrire !)



- Ecrire un algorithme utilisant des variables de type chaîne de caractères, et affichant quatre variantes possibles de cette phrase « belle journée, ciel ensoleillé avec une température douce ». On ne se soucie pas de la ponctuation, ni des majuscules.

- **Variables** t1, t2, t3, t4 **en Caractère**
  **Début**
  t1 ← " belle journée, "
  t2 ← " ciel ensoleillé "
  t3 ← " avec une température "
  t4 ← « douce"
  **Ecrire** t1 & " " & t2 & " " & t3 & " " & t4
  **Ecrire** t3 & " " & t2 & " " & t4 & " " & t1
  **Ecrire** t2 & " " & t3 & " " & t1 & " " & t4
  **Ecrire** t4 & " " & t1 & " " & t2 & " " & t3
  **Fin**

# Les Tests ou structures alternatives



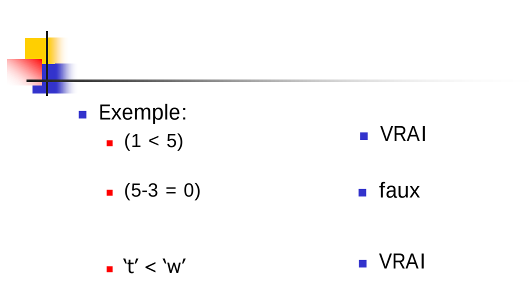
- Il n'y a que **deux formes possibles** pour un test

**Si** booléen **Alors**
  Instructions
**Finsi**

**Si** booléen **Alors**
  Instructions 1
**Sinon**
  Instructions 2
**Finsi**



- Un **booléen** est une **expression** dont la valeur est VRAI ou FAUX. Cela peut donc être (il n'y a que deux possibilités) :
  - une **variable** de type booléen
  - une **condition**



- Exemple:
  - (1 < 5) — VRAI
  - (5-3 = 0) — faux
  - 't' < 'w' — VRAI

## Qu'est ce qu'une condition ?



- Une condition est une comparaison
  - Une expression définit à partir d'un opérateur de comparaison
    - égal à…
    - différent de…
    - strictement plus petit que…
    - strictement plus grand que…
    - plus petit ou égal à…
    - plus grand ou égal à…
- L'évaluation d'une condition est un booléen

## Exercice



- Ecrire un algorithme qui demande un nombre à l'utilisateur, et l'informe ensuite si ce nombre est positif ou négatif (on laisse de côté le cas où le nombre vaut zéro).

**Variable** n **en Entier**
**Début**
**Ecrire** "Entrez un nombre : "
**Lire** n
**Si** n > 0 **Alors**
**Ecrire** "Ce nombre est positif"
**Sinon**
**Ecrire** "Ce nombre est négatif"
**Finsi**
Fin



## Conditions composées

- Certains problèmes exigent parfois de formuler des conditions qui ne peuvent pas être exprimées sous la forme simple
  - Composition de conditions.
    - Opérateurs de compositions:
      - connecteur logique: ET, OU, NON…
    - **Attention aux conditions qui ne pourront jamais être vraie, ou jamais être fausse**



- Ecrire un algorithme qui demande deux nombres à l'utilisateur et l'informe ensuite si leur produit est négatif ou positif (on laisse de côté le cas où le produit est nul). Attention toutefois : on ne doit **pas** calculer le produit des deux nombres.



- **Variables** m, n **en Entier**
  **Début**
  **Ecrire** "Entrez deux nombres : "
  **Lire** m, n
  **Si** (m > 0 ET n > 0) OU (m < 0 ET n < 0)
  **Alors**
  **Ecrire** "Leur produit est positif"
  **Sinon**
  **Ecrire** "Leur produit est négatif"
  **Finsi**
  **Fin**



- Ecrire un algorithme qui demande trois noms à l'utilisateur et l'informe ensuite s'ils sont rangés ou non dans l'ordre alphabétique.

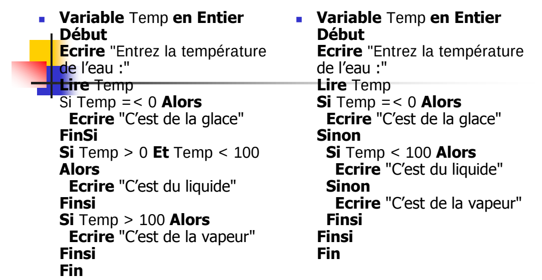
**Variables** a, b, c **en Caractère**
**Début**
**Ecrire** "Entrez successivement trois noms : "
**Lire** a, b, c
**Si** a < b ET b < c **Alors**
**Ecrire** "Ces noms sont classés alphabétiquement"
**Sinon**
**Ecrire** "Ces noms ne sont pas classés"
**Finsi**
**Fin**



## Tests imbriqués

- Les structures de tests imbriqués sont indispensables à la simplification et à l'optimisation des algorithmes.

- Une instruction peut être un test et ainsi de suite



```
Variable Temp en Entier
Début
Ecrire "Entrez la température
de l'eau :"
Lire Temp
Si Temp =< 0 Alors
  Ecrire "C'est de la glace"
FinSi
Si Temp > 0 Et Temp < 100
Alors
  Ecrire "C'est du liquide"
Finsi
Si Temp > 100 Alors
  Ecrire "C'est de la vapeur"
Finsi
Fin
```

```
Variable Temp en Entier
Début
Ecrire "Entrez la température
de l'eau :"
Lire Temp
Si Temp =< 0 Alors
  Ecrire "C'est de la glace"
Sinon
  Si Temp < 100 Alors
    Ecrire "C'est du liquide"
  Sinon
    Ecrire "C'est de la vapeur"
  Finsi
Finsi
Fin
```

•Lequel de ces deux programmes et le plus performants à l'exécution



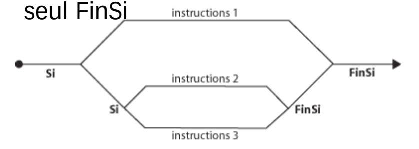
- Ecrire un algorithme qui demande un nombre à l'utilisateur, et l'informe ensuite si ce nombre est positif ou négatif (on inclut cette fois le traitement du cas où le nombre vaut zéro).



```
Variable n en Entier
Début
Ecrire "Entrez un nombre : "
Lire n
Si n < 0 Alors
  Ecrire "Ce nombre est négatif"
SinonSi n = 0 Alors
  Ecrire "Ce nombre est nul"
Sinon
  Ecrire "Ce nombre est positif"
Finsi
Fin
```



- Dans le cas de tests imbriqués, le Sinon et le Si peuvent être fusionnés en un SinonSi. On considère alors qu'il s'agit d'un seul bloc de test, conclu par un seul FinSi






- Ecrire un algorithme qui demande deux nombres à l'utilisateur et l'informe ensuite si le produit est négatif ou positif (on inclut cette fois le traitement du cas où le produit peut être nul). Attention toutefois, on ne doit pas calculer le produit !

- **Variables** m, n **en Entier**
  **Début**
  **Ecrire** "Entrez deux nombres : "
  **Lire** m, n
  **Si** m = 0 OU n = 0 **Alors**
  **Ecrire** "Le produit est nul"
  **SinonSi** (m < 0 ET n < 0) OU (m > 0 ET n > 0)
  **Alors**
  **Ecrire** "Le produit est positif"
  **Sinon**
  **Ecrire** "Le produit est négatif"
  **Finsi**
  **Fin**



- Ecrire un algorithme qui demande l'âge d'un enfant à l'utilisateur. Ensuite, il l'informe de sa catégorie :
- "Poussin" de 6 à 7 ans
- "Pupille" de 8 à 9 ans
- "Minime" de 10 à 11 ans
- "Cadet" après 12 ans
- Peut-on concevoir plusieurs algorithmes équivalents menant à ce résultat ?

- **Variable** age **en Entier**
  **Début**
  **Ecrire** "Entrez l'âge de l'enfant : "
  **Lire** age
  **Si** age >= 12 **Alors**
  **Ecrire** "Catégorie Cadet"
  **SinonSi** age >= 10 Alors
  **Ecrire** "Catégorie Minime"
  **SinonSi** age >= 8 Alors
  **Ecrire** "Catégorie Pupille"
  **SinonSi** age >= 6 Alors
  **Ecrire** "Catégorie Poussin"
  **Finsi**
  **Fin**
- On peut évidemment écrire cet algorithme de différentes façons, ne serait-ce qu'en commençant par la catégorie la plus jeune.



## Encore de la Logique

- **Dans une condition composée employant à la fois des opérateurs ET et des opérateurs OU, la présence de parenthèses possède une influence sur le résultat, tout comme dans le cas d'une expression numérique comportant des opérateurs numériques (exemple multiplications et des additions)**

- **Variables** A, B, C, D, E **en Booléen**
  **Variable** X **en Entier**
  **Début**
  **Lire** X
  A ← X > 12
  B ← X > 2
  C ← X < 6
  D ← (A ET B) OU C
  E ← A ET (B OU C)
  **Ecrire** D, E
  **Fin**

•Que vaut D et E?



- **Toute structure de test requérant une condition composée faisant intervenir l'opérateur ET peut être exprimée de manière équivalente avec un opérateur OU, et réciproquement.**
  - **Loi de Shannon/Morgan**



- **Si** A ET B **Alors**
  Instructions 1
  **Sinon**
  Instructions 2
  **Finsi**

  équivaut à :

  **Si** NON A OU NON B **Alors**
  Instructions 2
  **Sinon**
  Instructions 1
  **Finsi**



- Formulez un algorithme équivalent à l'algorithme suivant en permutant les **ET** et les **OU**:
- **Si** Tutu > Toto + 4 OU Tata = "OK"
  **Alors**
  Tutu ← Tutu + 1
  **Sinon**
  Tutu ← Tutu – 1
  **Finsi**



- il suffit d'appliquer la règle de la transformation du OU en ET vue en cours (loi de Morgan). Attention toutefois à la rigueur dans la transformation des conditions en leur contraire...
- **Si** Tutu <= Toto + 4 ET Tata <> "OK" **Alors**
  Tutu ← Tutu - 1
  **Sinon**
  Tutu ← Tutu + 1
  **Finsi**



- Cet algorithme est destiné à prédire l'avenir, et il doit être infaillible !
- Il lira au clavier l'heure et les minutes, et il affichera l'heure qu'il sera une minute plus tard. Par exemple, si l'utilisateur tape 21 puis 32, l'algorithme doit répondre :
- "Dans une minute, il sera 21 heure(s) 33".
- NB : on suppose que l'utilisateur entre une heure valide. Pas besoin donc de la vérifier.



- **Variables** h, m **en Numérique**
  **Début**
  **Ecrire** "Entrez les heures, puis les minutes : "
  **Lire** h, m
  m ← m + 1
  **Si** m = 60 **Alors**
  m ← 0
  h ← h + 1
  **FinSi**
  **Si** h = 24 **Alors**
  h ← 0
  **FinSi**
  **Ecrire** "Dans une minute il sera ", h, "heure(s) ", m, "minute(s)"
  **Fin**



- De même que le précédent, cet algorithme doit demander une heure et en afficher une autre. Mais cette fois, il doit gérer également les secondes, et afficher l'heure qu'il sera une seconde plus tard.
- Par exemple, si l'utilisateur tape 21, puis 32, puis 8, l'algorithme doit répondre : "Dans une seconde, il sera 21 heure(s), 32 minute(s) et 9 seconde(s)".
- NB : là encore, on suppose que l'utilisateur entre une date valide.

- **Variables** h, m, s **en Numérique**
**Début**
**Ecrire** "Entrez les heures, puis les minutes, puis les secondes : "
**Lire** h, m, s
s ← s + 1
**Si** s = 60 **Alors**
  s ← 0
  m ← m + 1
**FinSi**
**Si** m = 60 **Alors**
  m ← 0
  h ← h + 1
**FinSi**
**Si** h = 24 **Alors**
  h ← 0
**FinSi**
**Ecrire** "Dans une seconde il sera ", h, "h", m, "m et ", s, "s"
**Fin**



- Un magasin de reprographie facture 0,10 E les dix premières photocopies, 0,09 E les vingt suivantes et 0,08 E au-delà. Ecrivez un algorithme qui demande à l'utilisateur le nombre de photocopies effectuées et qui affiche la facture correspondante.



- **Variables** n, p **en Numérique**
**Début**
**Ecrire** "Nombre de photocopies : "
**Lire** n
**Si** n <= 10 **Alors**
  p ← n * 0,1
**SinonSi** n <= 30 **Alors**
  p ← 10 * 0,1 + (n – 10) * 0,09
**Sinon**
  p ← 10 * 0,1 + 20 * 0,09 + (n – 30) * 0,08
**FinSi**
**Ecrire** "Le prix total est: ", p
**Fin**



- Les habitants de Zorglub paient l'impôt selon les règles suivantes :
- les hommes de plus de 20 ans paient l'impôt
- les femmes paient l'impôt si elles ont entre 18 et 35 ans
- les autres ne paient pas d'impôt
- Le programme demandera donc l'âge et le sexe du Zorglubien, et se prononcera donc ensuite sur le fait que l'habitant est imposable.

- **Variable** sex **en Caractère**
**Variable** age **en Numérique**
**Début**
**Ecrire** "Entrez le sexe (M/F) : "
**Lire** sex
**Ecrire** "Entrez l'âge: "
**Lire** age
C1 ← sex = "M" ET age > 20
C2 ← sex = "F" ET (age > 18 ET age < 35)
**Si** C1 ou C2 **Alors**
  **Ecrire** "Imposable"
**Sinon**
  **Ecrire** "Non Imposable"
**FinSi**
**Fin**

## Au-delà de la logique : le style



- **De manière générale, il n'y a jamais une seule manière juste de traiter un problème. Entre les différentes possibilités, qui ne sont parfois pas meilleures les unes que les autres, le choix est une affaire de style.**



- **Dans une structure alternative complexe, les conditions composées, l'imbrication des structures de tests et l'emploi des variables booléennes ouvrent la possibilité de choix stylistiques différents.**
  - **L'alourdissement des conditions allège les structures de tests et le nombre des booléens nécessaires ;**
  - **l'emploi de booléens supplémentaires permet d'alléger les conditions et les structures de tests, et ainsi de suite.**

- Les élections législatives, en Guignolerie Septentrionale, obéissent à la règle suivante :
  - lorsque l'un des candidats obtient plus de 50% des suffrages, il est élu dès le premier tour.
  - en cas de deuxième tour, peuvent participer uniquement les candidats ayant obtenu au moins 12,5% des voix au premier tour.

- Vous devez écrire un algorithme qui permette la saisie des scores de quatre candidats au premier tour. Cet algorithme traitera ensuite le candidat numéro 1 (et **uniquement** lui) : il dira s'il est élu, battu, s'il se trouve en ballottage favorable (il participe au second tour en étant arrivé en tête à l'issue du premier tour) ou défavorable (il participe au second tour sans avoir été en tête au premier tour).

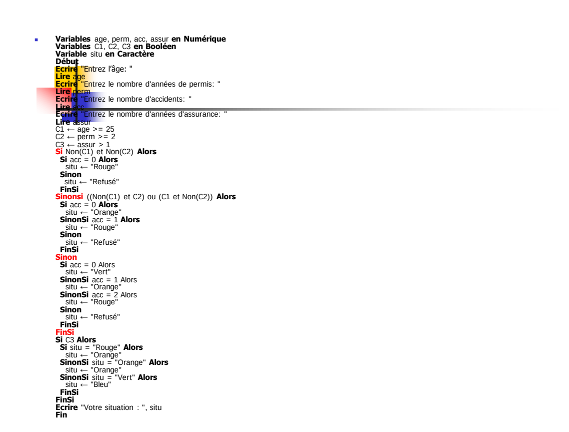
```
Variables A, B, C, D en Numérique
Début
Ecrire "Entrez les scores des quatre prétendants :"
Lire A, B, C, D
C1 ← A > 50
C2 ← B > 50 ou C > 50 ou D > 50
C3 ← A >= B et A >= C et A >= D
C4 ← A >= 12,5
Si C1 Alors
  Ecrire "Elu au premier tour"
Sinonsi C2 ou Non(C4) Alors
  Ecrire "Battu, éliminé, sorti !!!"
SinonSi C3 Alors
  Ecrire "Ballotage favorable"
Sinon
  Ecrire "Ballotage défavorable"
FinSi
Fin
```

- **si le candidat a plus de 50%, il est élu, sinon s'il a plus de 12,5 %, il est au deuxième tour, sinon il est éliminé. Hé hé hé... mais il ne faut pas oublier que le candidat peut très bien avoir eu 20 % mais être tout de même éliminé, tout simplement parce que l'un des autres a fait plus de 50 % et donc qu'il n'y a pas de deuxième tour !...**
  - **Moralité : ne jamais se jeter sur la programmation avant d'avoir soigneusement mené l'analyse du problème à traiter.**

- Une compagnie d'assurance automobile propose à ses clients quatre familles de tarifs identifiables par une couleur, du moins au plus onéreux : tarifs bleu, vert, orange et rouge. Le tarif dépend de la situation du conducteur :
  - un conducteur de moins de 25 ans et titulaire du permis depuis moins de deux ans, se voit attribuer le tarif rouge, si toutefois il n'a jamais été responsable d'accident. Sinon, la compagnie refuse de l'assurer.
  - un conducteur de moins de 25 ans et titulaire du permis depuis plus de deux ans, ou de plus de 25 ans mais titulaire du permis depuis moins de deux ans a le droit au tarif orange s'il n'a jamais provoqué d'accident, au tarif rouge pour un accident, sinon il est refusé.
  - un conducteur de plus de 25 ans titulaire du permis depuis plus de deux ans bénéficie du tarif vert s'il n'est à l'origine d'aucun accident et du tarif orange pour un accident, du tarif rouge pour deux accidents, et refusé au-delà
  - De plus, pour encourager la fidélité des clients acceptés, la compagnie propose un contrat de la couleur immédiatement la plus avantageuse s'il est entré dans la maison depuis plus d'un an.
  - Ecrire l'algorithme permettant de saisir les données nécessaires (sans contrôle de saisie) et de traiter ce problème. Avant de se lancer à corps perdu dans cet exercice, on pourra réfléchir un peu et s'apercevoir qu'ils est plus simple qu'il en a l'air (cela s'appelle faire une analyse !)



- deux corrigés différents. Le premier suit l'énoncé pas à pas. C'est juste, mais c'est vraiment lourd.



- Reprenons juste après l'affectation des trois variables booléennes C1, C2, et C3. On écrit :

```
P ← 0
Si Non(C1) Alors
  P ← P + 1
FinSi
Si Non(C2) Alors
  P ← P + 1
FinSi
P ← P + acc
Si P < 3 et C3 Alors
  P ← P - 1
FinSi
Si P = -1 Alors
  situ ← "Bleu"
SinonSi P = 0 Alors
  situ ← "Vert"
SinonSi P = 1 Alors
  situ ← "Orange"
SinonSi P = 2 Alors
  situ ← "Rouge"
Sinon
  situ ← "Refusé"
FinSi
Ecrire "Votre situation : ", situ
Fin
```